Milepost Based Addressing Tools

Contributed by Bert Granberg
30, Oct. 2007
Last Updated 30, Oct. 2007

In some cases, rural roads are addressed using a mile post-based addressing scheme. These scripts provide some basic utility for assigning mile post-based address ranges to road features. The basic workflow is to select one feature of the road to be addressed with milepost values. Then expand the selection wither manually or using one of the scripts below to include all the road features to be assigned address ranges. Then run the renumber script (details below).

This VBA Code for ArcMap includes three sub procedures that for ease of use should be linked to UIButtonControls on a new toolbar.
- SelectConnected_LikeName(). Expands the current selection from one road feature to include features with the same name that are simply connected. Will not recursively select all features if spurs or branching is present. A suggestion for the button name is: Expand Connected
- Public Sub ReNumberSelected_LikeName(). Will prompt user for a starting milepost value, a starting position from which to renumber the selected feature, and the option to flip, where necessary, the selected features so their orientation is consistent with increasing address range values. Suggested button name: Renumber
- A script to clear all graphics present in the current data frame map view. Suggested button name: Clear Graphics

It is also suggested that the basic Select Features tool be added to the new toolbar so it looks something like this:

Here's the code:

```
Option Explicit

Public Sub SelectConnected_LikeName()

  'Given one selected feature in the roads layer, this script selects connected roads
  'with the same name so that the selected set can be made into a single part, non
  'branching feature using the RenumberSelected_LikeName script

  'Get reference to ArcMap document
  Dim pMxDoc As IMxDocument
  Dim pMap As IMap
  Dim pActiveView As IActiveView
  Set pMxDoc = ThisDocument
  Set pMap = pMxDoc.FocusMap
  Set pActiveView = pMap

  'Get reference to road layer objects
  Dim roadLayerName As String
  Dim roadSName_FieldName As String
  Dim road_LFFieldName As String
  Dim road_RFFieldName As String
  Dim road_LTFieldName As String
  Dim road_RTFieldName As String
  Dim roadSType_FieldName As String
  Dim roadSufDir_FieldName As String

  Dim roadSNAME As String
  Dim roadSType As String
  Dim roadSufDir As String
  Dim classQueryStr As String

  Dim pRoadFL As IFeatureLayer
  Dim pRoadFC As IFeatureClass
  Dim pRoadFSel As IFeatureSelection
```

```
Dim pRoadSelSet As ISelectionSet
Dim use_stype_sufdir_constraint As Boolean

'*******  USER DEFINED SETTINGS - BEGIN *******
'***SET ROAD LAYER NAME HERE
roadLayerName = "SanJuanRoads"
'***SET ROAD NAME FIELD HERE
roadSName_FieldName = "S_NAME"
'***SET CONSTRAINT HERE (for use when roadSName_FieldName = S_NAME)
'   prevents selection of wrong segments at intersections where numeric street names are the same
use_stype_sufdir_constraint = True
'***SET CLASS QUERY STR HERE
classQueryStr = ""
'classQueryStr = " and CLASS = 'B'"

'*******  USER DEFINED SETTINGS - END *******

roadSType_FieldName = "S_TYPE"
roadSufDir_FieldName = "SUF_DIR"

Dim roadSName_FieldIndex  As Integer
Dim roadSType_FieldIndex  As Integer
Dim roadSufDir_FieldIndex  As Integer

Dim pEnumLayer As IEnumLayer
Dim pLayer As IFeatureLayer
Dim pQF As IQueryFilter
Dim pFeatureCursor As IFeatureCursor
Dim pFeature As IFeature

Dim pPrevfeature As IFeature

Dim pSelFeature As IFeature
Dim pCurrFeature As IFeature
Dim pEndpt As IPoint
Dim pStartPt As IPoint
Dim currPolyline As IPolyline
Dim startFound As Boolean
Dim endFound As Boolean
Dim pSpatialFilter As ISpatialFilter
Dim pTempFCursor As IFeatureCursor
Dim pTempFeature As IFeature
Dim pTempPolyline As IPolyline
Dim pTempPoint As IPoint
Dim pTempRelOP As IRelationalOperator

Set pEnumLayer = pMap.Layers
Set pLayer = pEnumLayer.Next


Do Until pLayer Is Nothing
   Debug.Print pLayer.Name

   If pLayer.Name = roadLayerName Then
      startFound = False
      endFound = False
      Set pRoadFSel = pLayer
      Set pRoadFL = pLayer
      Set pRoadFC = pRoadFL.FeatureClass
      Set pRoadSelSet = pRoadFSel.SelectionSet

      If pRoadSelSet.count = 1 Then
         Debug.Print pRoadSelSet.IDs.Next
         Set pRoadFL = pLayer
         Set pRoadFC = pRoadFL.FeatureClass
```

```
        roadSName_FieldIndex = pRoadFC.FindField(roadSName_FieldName)
        roadSType_FieldIndex = pRoadFC.FindField(roadSType_FieldName)
        roadSufDir_FieldIndex = pRoadFC.FindField(roadSufDir_FieldName)

        If Not roadSName_FieldIndex >= 0 Then
            MsgBox "Road name field not found...exiting"
            Exit Sub
        End If

        If use_stype_sufdir_constraint Then
            If Not roadSType_FieldIndex >= 0 Then
                MsgBox "Road type field not found...exiting"
                Exit Sub
            End If
            If Not roadSufDir_FieldIndex >= 0 Then
                MsgBox "Road suffix direction field not found...exiting"
                Exit Sub
            End If
        End If

        Set pSelFeature = pRoadFC.GetFeature(pRoadSelSet.IDs.Next)
        Set pCurrFeature = pSelFeature

        roadSNAME = pCurrFeature.Value(roadSName_FieldIndex)

        If Not IsNull(pCurrFeature.Value(roadSType_FieldIndex)) Then
           roadSType = Replace(pCurrFeature.Value(roadSType_FieldIndex), " ", "")
        Else
           roadSType = ""
        End If

        If Not IsNull(pCurrFeature.Value(roadSufDir_FieldIndex)) Then
           roadSufDir = Replace(pCurrFeature.Value(roadSufDir_FieldIndex), " ", "")
        Else
           roadSufDir = ""
        End If


        Set currPolyline = pCurrFeature.Shape
        Set pStartPt = currPolyline.FromPoint

        Do Until startFound = True

            Set currPolyline = pCurrFeature.Shape

            Set pSpatialFilter = New SpatialFilter
            With pSpatialFilter
               Set .Geometry = pStartPt

               If pPrevfeature Is Nothing Then
               .WhereClause = roadSName_FieldName & " = '" & roadSNAME & "' and " & pRoadFC.OIDFieldName & " <> " &
pCurrFeature.OID & classQueryStr
               Else
               .WhereClause = roadSName_FieldName & " = '" & roadSNAME & "' and " & pRoadFC.OIDFieldName & " <> " &
pCurrFeature.OID & " and " & pRoadFC.OIDFieldName & " <> " & pPrevfeature.OID & classQueryStr
               End If
               .SpatialRel = esriSpatialRelIntersects

            End With

            If use_stype_sufdir_constraint Then
               If roadSType <> "" Then
                  pSpatialFilter.WhereClause = pSpatialFilter.WhereClause & " and " & roadSType_FieldName & " = '" & roadSTy
& classQueryStr
```

```
                End If
                If roadSufDir <> "" Then
                    pSpatialFilter.WhereClause = pSpatialFilter.WhereClause & " and " & roadSufDir_FieldName & " = '" & roadSufI
& classQueryStr
                End If
            End If

            Debug.Print pSpatialFilter.WhereClause

            Set pTempFCursor = pRoadFL.Search(pSpatialFilter, True)
            Set pTempFeature = pTempFCursor.NextFeature

            If Not pTempFeature Is Nothing Then
                Debug.Print pTempFeature.OID
                If pTempFeature.OID = pCurrFeature.OID Then
                    Set pTempFeature = pTempFCursor.NextFeature
                    If pTempFeature Is Nothing Then
                        startFound = True
                        Exit Do
                    End If
                End If
                Set pTempPolyline = pTempFeature.Shape
                Set pTempPoint = pTempPolyline.FromPoint
                Set pTempRelOP = pTempPoint
                If pTempRelOP.Equals(pStartPt) Then
                    'use topoint instead
                    pRoadSelSet.Add (pTempFeature.OID)
                    Set pStartPt = pTempPolyline.ToPoint
                Else
                    'use frompoint
                    pRoadSelSet.Add (pTempFeature.OID)
                    Set pStartPt = pTempPolyline.FromPoint
                End If
            Else
                startFound = True
            End If

            Set pPrevfeature = pCurrFeature
            Set pCurrFeature = pTempFeature


        Loop

        Set pCurrFeature = pSelFeature
        Set currPolyline = pCurrFeature.Shape

        Set pEndpt = currPolyline.ToPoint

        Do Until endFound = True

            Set pSpatialFilter = New SpatialFilter
            With pSpatialFilter
                Set .Geometry = pEndpt

                If pPrevfeature Is Nothing Then
                    .WhereClause = roadSName_FieldName & " = '" & roadSNAME & "' and " & pRoadFC.OIDFieldName & " <> " &
pCurrFeature.OID & classQueryStr
                Else
                    .WhereClause = roadSName_FieldName & " = '" & roadSNAME & "' and " & pRoadFC.OIDFieldName & " <> " &
pCurrFeature.OID & " and " & pRoadFC.OIDFieldName & " <> " & pPrevfeature.OID & classQueryStr
                End If

                .SpatialRel = esriSpatialRelIntersects
            End With
```

```
            If use_stype_sufdir_constraint Then
               If roadSType <> "" Then
                  pSpatialFilter.WhereClause = pSpatialFilter.WhereClause & " and " & roadSType_FieldName & " = '" & roadSTy
& classQueryStr
               End If
               If roadSufDir <> "" Then
                  pSpatialFilter.WhereClause = pSpatialFilter.WhereClause & " and " & roadSufDir_FieldName & " = '" & roadSufI
& classQueryStr
               End If
            End If

            Debug.Print pSpatialFilter.WhereClause

            Set pTempFCursor = pRoadFL.Search(pSpatialFilter, True)
            Set pTempFeature = pTempFCursor.NextFeature

            If Not pTempFeature Is Nothing Then
               'If pTempFeature.OID = 6738 Then
               '    Debug.Print "here"
               'End If
               Debug.Print pTempFeature.OID
               If pTempFeature.OID = pCurrFeature.OID Then
                  If pTempFeature Is Nothing Then
                     endFound = True
                     Exit Do
                  Else
                     Set pTempFeature = pTempFCursor.NextFeature
                  End If
               End If
               Set pTempPolyline = pTempFeature.Shape
               Set pTempPoint = pTempPolyline.FromPoint
               Set pTempRelOP = pTempPoint
               If pTempRelOP.Equals(pEndpt) Then
                  'use topoint instead
                  pRoadSelSet.Add (pTempFeature.OID)
                  Set pEndpt = pTempPolyline.ToPoint
               Else
                  'use frompoint
                  pRoadSelSet.Add (pTempFeature.OID)
                  Set pEndpt = pTempPolyline.FromPoint
               End If
            Else
               endFound = True
            End If

            Set pPrevfeature = pCurrFeature
            Set pCurrFeature = pTempFeature


         Loop

         Set pRoadFSel.SelectionSet = pRoadSelSet

      Else
         MsgBox "this tool works when only one feature is selected in the " & roadLayerName & " layer"
      End If


      Exit Do
   End If
   Set pLayer = pEnumLayer.Next
 Loop

 pActiveView.Refresh
End Sub
```

```
Public Sub ReNumberSelected_LikeName()

  'Given a selection of connected, non-branching, non-looping features in the roads layer
  'this script populates the address range fields with milepost values given user input for
  'starting point, beginning milepost value, and whether or non to reorient features to agree
  'with the milepost numbering direction.

  'GET REFERENCE TO CURRENT MAP DOCUMENT
  Dim pMxDoc As IMxDocument
  Dim pMap As IMap
  Dim pActiveView As IActiveView
  Set pMxDoc = ThisDocument
  Set pMap = pMxDoc.FocusMap
  Set pActiveView = pMap

  Dim roadLayerName As String
  Dim roadSName_FieldName As String
  Dim road_LFFieldName As String
  Dim road_RFFieldName As String
  Dim road_LTFieldName As String
  Dim road_RTFieldName As String

  '*******  USER DEFINED SETTINGS - BEGIN *******
  '*** SET THIS VARIABLE TO YOUR ROAD LAYER'S NAME IN ARCMAP
  roadLayerName = "SanJuanRoads"

  '*** SET FIELDS TO RECEIVE MILEPOST NUMBERING VALUES
  road_LFFieldName = "L_F_ADD"
  road_RFFieldName = "R_F_ADD"
  road_LTFieldName = "L_T_ADD"
  road_RTFieldName = "R_T_ADD"

  'user prompting for inputs is also used in this script below
  '*******  USER DEFINED SETTINGS - END *******

  'DIMENSION VARIABLE USED IN ANALYSIS
  Dim pEnumLayer As IEnumLayer
  Dim pLayer As ILayer
  Dim pFLayer As IFeatureLayer
  Dim pQF As IQueryFilter
  Dim pFeatureCursor As IFeatureCursor
  Dim pFeature As IFeature

  Dim pRoadFL As IFeatureLayer
  Dim pRoadFC As IFeatureClass
  Dim pRoadFSel As IFeatureSelection
  Dim pRoadSelSet As ISelectionSet

  Dim road_LFFieldIndex As Integer
  Dim road_RFFieldIndex  As Integer
  Dim road_LTFieldIndex  As Integer
  Dim road_RTFieldIndex  As Integer

  Dim pEndpt As IPoint
  Dim currPolyline As IPolyline
  Dim pGeometryCollection As iGeometryCollection
  Dim endFound As Boolean

  Dim pSpatialFilter As ISpatialFilter
  Dim pCurrFeature As IFeature
  Dim pUnionPolylineTopOp As ITopologicalOperator
  Dim firstFeature As Boolean
  Dim currGeometry As IGeometry
```

```
Dim pUnionPolyline As IPolyline

Dim pFCursor As IFeatureCursor
Dim pCurve As ICurve

Dim pGraphicsContainer As IGraphicsContainer
Dim pElement As IElement
Dim pMarkerElement As IMarkerElement
Dim pMarkerSymbol As ISimpleMarkerSymbol
Dim pColor As IRgbColor, pLineColor As IRgbColor, pPolyFillColor As IColor

firstFeature = True

Set pEnumLayer = pMap.Layers
Set pLayer = pEnumLayer.Next

Dim roadLayerFound As Boolean
roadLayerFound = False

'LOOP THROUGH LAYERS UNTIL THE ROAD LAYER IS FOUND
Do Until pLayer Is Nothing Or roadLayerFound

   If pLayer.Name = roadLayerName Then
      If TypeOf pLayer Is IFeatureLayer Then

         roadLayerFound = True
         Set pFLayer = pLayer
         Set pRoadFSel = pFLayer
         Set pRoadFL = pFLayer
         Set pRoadFC = pRoadFL.FeatureClass
         Set pRoadSelSet = pRoadFSel.SelectionSet

         road_LFFieldIndex = pRoadFC.FindField(road_LFFieldName)
         road_RFFieldIndex = pRoadFC.FindField(road_RFFieldName)
         road_LTFieldIndex = pRoadFC.FindField(road_LTFieldName)
         road_RTFieldIndex = pRoadFC.FindField(road_RTFieldName)

         If road_LFFieldIndex = -1 Or road_RFFieldIndex = -1 Or road_LTFieldIndex = -1 Or road_RTFieldIndex = -1 Then
            MsgBox "At least one address range field is missing...Exiting"
            Exit Sub
         End If

         pRoadSelSet.Search Nothing, True, pFCursor
         Set pCurrFeature = pFCursor.NextFeature

         'Union all selected polylines into one polyline feature
         Do While Not pCurrFeature Is Nothing
            Debug.Print pCurrFeature.OID
            Set currGeometry = pCurrFeature.ShapeCopy

            'currGeometry.Project pMap.SpatialReference

            If firstFeature Then
               Set pUnionPolyline = currGeometry
               firstFeature = False
            Else
               Set pUnionPolylineTopOp = pUnionPolyline 'QI
               pUnionPolylineTopOp.Simplify
               Set pUnionPolyline = pUnionPolylineTopOp.Union(currGeometry)
            End If

            Set pCurrFeature = pFCursor.NextFeature

         Loop
```

```
Set pUnionPolylineTopOp = pUnionPolyline 'QI
pUnionPolylineTopOp.Simplify

Set pGeometryCollection = pUnionPolyline

If pGeometryCollection.GeometryCount = 1 Then

   'SYMBOLIZE MAP WITH GREEN CIRCLE AND RED BOX

   Set pCurve = pUnionPolylineTopOp
   pCurve.Project pMap.SpatialReference

   Set pColor = New RgbColor
   pColor.Green = 153
   Set pMarkerSymbol = New SimpleMarkerSymbol
   pMarkerSymbol.Color = pColor
   pMarkerSymbol.size = 10
   pMarkerSymbol.Style = esriSMSCircle

   Set pElement = New MarkerElement
   pElement.Geometry = pCurve.FromPoint
   Set pMarkerElement = pElement 'QI
   pMarkerElement.Symbol = pMarkerSymbol
   Set pGraphicsContainer = pActiveView.GraphicsContainer
   pGraphicsContainer.AddElement pElement, 0

   Set pColor = New RgbColor
   pColor.Red = 255
   Set pMarkerSymbol = New SimpleMarkerSymbol
   pMarkerSymbol.Color = pColor
   pMarkerSymbol.size = 8
   pMarkerSymbol.Style = esriSMSSquare

   Set pElement = New MarkerElement
   pElement.Geometry = pCurve.ToPoint
   Set pMarkerElement = pElement 'QI
   pMarkerElement.Symbol = pMarkerSymbol
   Set pGraphicsContainer = pMxDoc.ActiveView.GraphicsContainer
   pGraphicsContainer.AddElement pElement, 0

   pActiveView.Refresh

   '*******  USER PROMPTING VALUES - BEGIN *******

   Dim msgboxResponse As Long
   msgboxResponse = MsgBox("Start milepost numbering from Green Circle??" & vbNewLine & vbNewLine & _
            "(clicking no will start the numbering from the red square)", _
            vbYesNoCancel, "Milepost Numbering Start Position?")

   Debug.Print msgboxResponse

   If msgboxResponse = 6 Then ' YES
      Set pEndpt = pCurve.FromPoint
   ElseIf msgboxResponse = 7 Then 'NO
      Set pEndpt = pCurve.ToPoint
   Else
      MsgBox "Exiting without renumbering", vbOKOnly, "Exit"
      Exit Sub ' CANCEL
   End If

   Dim BeginMilePostStr As String
   Dim BeginMilePostVal As Double
   BeginMilePostStr = "-1"

   Do Until CDbl(BeginMilePostStr) >= 0
```

```
            BeginMilePostStr = InputBox("Enter the begining milepost value", "Beginning milepost value?", 0)
            If BeginMilePostStr = "" Then
               'cancel or empty string encountered
               MsgBox "Exiting without renumbering", vbOKOnly, "Exit"
               Exit Sub
            End If
            If Not IsNumeric(BeginMilePostStr) Then
               'non-numeric value entered
               MsgBox "Value must be a non-negative number", vbOKOnly, "Invalid Milepost Value"
               BeginMilePostStr = "-1"
            End If
         Loop

         BeginMilePostVal = CDbl(BeginMilePostStr)

         Dim flipToMPDirection As Boolean
         msgboxResponse = MsgBox("Flip selected features direction to agree with milepost numbering direction??",
vbYesNoCancel, "Flip Feature Orientation?")

         If msgboxResponse = 6 Then 'YES
            flipToMPDirection = True
         ElseIf msgboxResponse = 7 Then 'NO
            flipToMPDirection = False
         Else 'CANCEL
            MsgBox "Exiting without renumbering", vbOKOnly, "Exit"
            Exit Sub
         End If

         '*******  USER PROMPTING VALUES - END *******

         Dim pTopOp As ITopologicalOperator
         Set pTopOp = pEndpt
         Dim pbuff As IPolygon
         Set pbuff = pTopOp.Buffer(1) '1 meter buffer

         Set pSpatialFilter = New SpatialFilter
         With pSpatialFilter
            Set .Geometry = pbuff
            .WhereClause = ""
            .SpatialRel = esriSpatialRelIntersects

         End With

         pRoadSelSet.Search pSpatialFilter, True, pFCursor
         Set pCurrFeature = pFCursor.NextFeature

         Do Until endFound = True

            Set currPolyline = pCurrFeature.Shape
            Debug.Print pCurrFeature.OID

            If flipToMPDirection Then
               If CLng(currPolyline.ToPoint.X * 10) = CLng(pEndpt.X * 10) And CLng(currPolyline.ToPoint.Y * 10) = CLng(pEn
10) Then
                  currPolyline.ReverseOrientation
               End If

               Set pCurrFeature.Shape = currPolyline

               'for next iteration
               Set pEndpt = currPolyline.ToPoint

            Else
               'for next iteration
               Set pEndpt = currPolyline.ToPoint
```

```
                End If

                With pCurrFeature

                    .Value(road_LFFieldIndex) = BeginMilePostVal
                    'will currently report milepost values to the thousandth of a mile
                    .Value(road_LTFieldIndex) = BeginMilePostVal + CInt((currPolyline.Length) / 1.609344) / 1000
                    .Value(road_RFFieldIndex) = BeginMilePostVal
                    'will currently report milepost values to the thousandth of a mile
                    .Value(road_RTFieldIndex) = BeginMilePostVal + CInt((currPolyline.Length) / 1.609344) / 1000
                    .Store

                End With

                'for next iteration
                BeginMilePostVal = BeginMilePostVal + CInt((currPolyline.Length) / 1.609344) / 1000

                Dim lastOID As Long
                lastOID = pCurrFeature.OID
                Set pSpatialFilter = New SpatialFilter
                With pSpatialFilter
                   Set .Geometry = pEndpt
                   .WhereClause = pRoadFC.OIDFieldName & " <> " & pCurrFeature.OID
                   .SpatialRel = esriSpatialRelIntersects

                End With

                pRoadSelSet.Search pSpatialFilter, True, pFCursor
                Set pCurrFeature = pFCursor.NextFeature

                If Not pCurrFeature Is Nothing Then
                  If pCurrFeature.OID = lastOID Then
                     Set pCurrFeature = pFCursor.NextFeature
                  End If
                Else
                  endFound = True
                End If

            Loop
            Dim runClearGraphics As Long
            runClearGraphics = MsgBox("Milepost numbering successful. Clear graphics now?", vbYesNo, "Milepost Numbering
Successful")

            If runClearGraphics = vbYes Then
               Call clearGraphics
            End If

        Else
            MsgBox "This function can only run on selected roads that form a simple, single-part geometry"
        End If



    End If

  End If

  Set pLayer = pEnumLayer.Next
Loop



End Sub

Public Sub clearGraphics()
```

```
  'If there are graphics in the map, this script deletes them and the view refreshed.
  Dim pMxDoc As IMxDocument
  Set pMxDoc = ThisDocument

  Dim count As Integer
  Dim pElement As IElement
  Dim pGraphicsContainer As IGraphicsContainer

  Set pGraphicsContainer = pMxDoc.ActiveView.GraphicsContainer

  pGraphicsContainer.Reset
  Set pElement = pGraphicsContainer.Next

  Do While Not pElement Is Nothing
    count = count + 1
    Set pElement = pGraphicsContainer.Next
  Loop

  If count > 0 Then
     pGraphicsContainer.DeleteAllElements
     pMxDoc.ActiveView.Refresh
  End If

End Sub
```